

ABSTRAK

Dalam pembuatan program aplikasi harus diperhatikan kecepatan waktu pelaksanaan program dan besarnya memori yang dipakai. Kecepatan suatu program dipengaruhi oleh penggunaan perintah-perintah dalam program, terutama perintah-perintah dalam *loop*. Sedangkan besarnya penggunaan memori dipengaruhi jumlah pemakaian variabel dalam suatu program. Ketidakefisienan dalam penggunaan perintah dan pemakaian variabel dalam suatu program menyebabkan unjuk kerja tidak optimal.

Ada beberapa hal yang ingin dicapai dalam mengerjakan Tugas Akhir ini yaitu mengimplementasikan teknik-teknik optimasi termasuk teknik optimasi *loop unrolling* dan *loop jamming* serta *strength reduction* pada *intermediate code* yang dihasilkan oleh kompilator berbahasa Indonesia, dan melakukan optimasi *intermediate code* program.

Optimasi *intermediate code* merupakan usaha untuk mengubah *intermediate code* menjadi lebih efisien, baik dilihat dari waktu pelaksanaan maupun besarnya pemakaian memori. *Intermediate code* merupakan suatu representasi internal program sumber.

Proses optimasi *intermediate code* terbagi atas optimasi lokal yaitu optimasi yang dijalankan pada suatu blok dasar dan optimasi global yaitu optimasi yang dijalankan pada keseluruhan blok program. Optimasi lokal meliputi pembentukan blok dasar, DAG (*Directed Acyclic Graph*), penyusunan ulang deretan perintah *quadruple*. Optimasi global meliputi pembentukan graf alir, eliminasi sub-ekspresi umum dan perambatan perintah penyalinan, pemindahan perintah invarian, *strength reduction* dan eliminasi variabel induksi, *loop jamming* serta *loop unrolling*.

Obyek optimasi *intermediate code* adalah *intermediate code* yang dibangkitkan kompilator bahasa pemrograman dengan grammar bahasa Indonesia. Input dari optimasi *intermediate code* adalah program sumber (*source program*), sedang hasil dari proses optimasi adalah *file intermediate code* yang sudah dioptimasi.